Project: Gender Classification from Blog Data with Machine Learning

Rasmus Eriksen, Rikke Randløv & Sofie Janas

April 8, 2016

Contents

1 Introduction

Classifying large amounts of data is increasingly important in the modern, digital society, and methods such as data mining allows one to extract data from diverse regions. An important tool in classifying data are various methods within machine learning, such as boosted decision trees and support vector machines. We will utilize these two tools for gender classification and prediction for a large dataset of internet blogs collected by the authors of [1].

2 Previous studies

Classification of various personal information (gender, age, personality type, political leanings) is a large and ongoing research topic within fields such as machine learning and linguistics. Previous studies we have read ([1], [2]) have utilized a large number of different predictors to determine the gender of the blog's author. The predictors includes various stylistic features (number of posts, number of characters in posts, posting frequencies etc.) and content-related features (special recurring words, occurences of verbs, adjectives, personal pronouns etc). Armed with these predictors, studies show accuracies of 62-89% with various methods [2].

3 Data set

The data set used in this project is 18.548 blogs, which contains, along the actual content of the blogs, also associated values of the bloggers' gender, age, occupation and zodiac. The blogs have been collected from the site blogger.com in august 2004. Each blog contains all posts the blogger has written up until the sampling date. We wish to use the classification tool to be able to consistently identify the gender of the blogger. It should be noted, that we only have data from bloggers in the age intervals: 13-17, 23-27 and 33-42.

When analysing the blogs with the intention of identifying the gender of the author, we need a set of predictors to run the learning algorithms on. Here we have two kinds of data: we have meta data, giving us the gender, age and job; and we have the actual blog, from which we can run a set of summary statistics on. The meta data itself is applicable to the learning algorithm with minor modifications: The genders "male" and "female" can be directly translated into the binary labels $(1, 2)$. The age does not need any translation and can be used directly. The jobs are pre-processed into 40 categories, and here each category can be assigned a label $(1, 2, \ldots, 40)$.
This gives us the first 2 predictors:

1) Age
2) Job

For summary statistics we choose a few simple statistics, and a few more elaborate, and we will later consider the effectiveness of these statistics. The summary statistics give 11 additional predictors.

3) The average post length (measured in number of characters)
4) The average word length (measured in number of characters)
5) The average posting frequency
6) The percentage of "Descriptors" (defined below)
7) The average number of URL links per post
8) The percentage of "Internet Words" (defined below)

9-13) The percentages of the words: {"the", "i", "him", "of", "and"}

In the above section "Internet Words" are defined to be from the following list: (lol, wtf, wth, dafuq, rofl, omg, brb, btw, irl, imo, jk, np, thx, imho, bae, bff, gr8, fml, l8r, yolo, swag, fyi, nvm, glhf, lulz, tbh, fail, 404, xoxo, stfu, 1337, noob, afaik, iou, asap, asl, w00t, cu, cya, ffs, fml, tmi, ftw, fu, derp, gtfo, haxor, idk, srsly, tldr, m8, sext, troll, sk8, sk8r, ty)
And "Descriptors" are defined as words ending with[2]: (able, al, ful, ible, ic, ive, less, ly, ous).

The distribution and correlations of the predictor data is shown in figure 1 as a corner plot: in the diagonal histograms of the 8 predictors are plotted, and in the sub-diagonal is shown the 2D correlation plots between parameter i and j. It is obvious that none of the 8 predictors seem to separate the data very well on their own.

We will also train the adaptively boosted decision tree using the predictors 1-13, with the last 5 predictors having a cornerplot as shown in figure 2

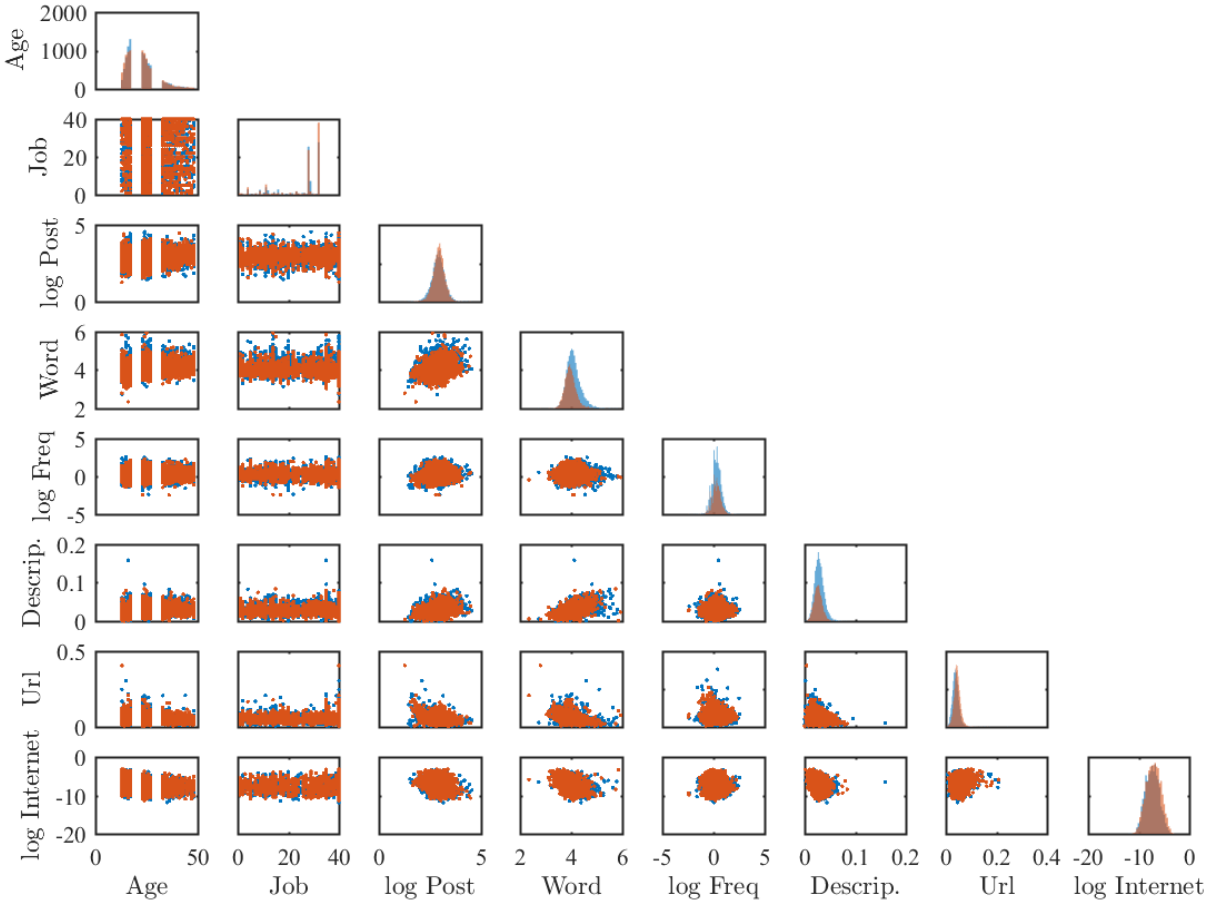


Figure 1: Corner plot of the predictors 1-8 from above. Red indicates female, and blue indicates male. The diagonal shows the histogram of each predictor, whereas the plot (i,j) is the correlation between predictor i and j. For the predictors average post length, frequency and percentage of "internet words" the logarithm of that predictor is shown instead for ease of view. (A larger plot of the figure can be seen after the references).

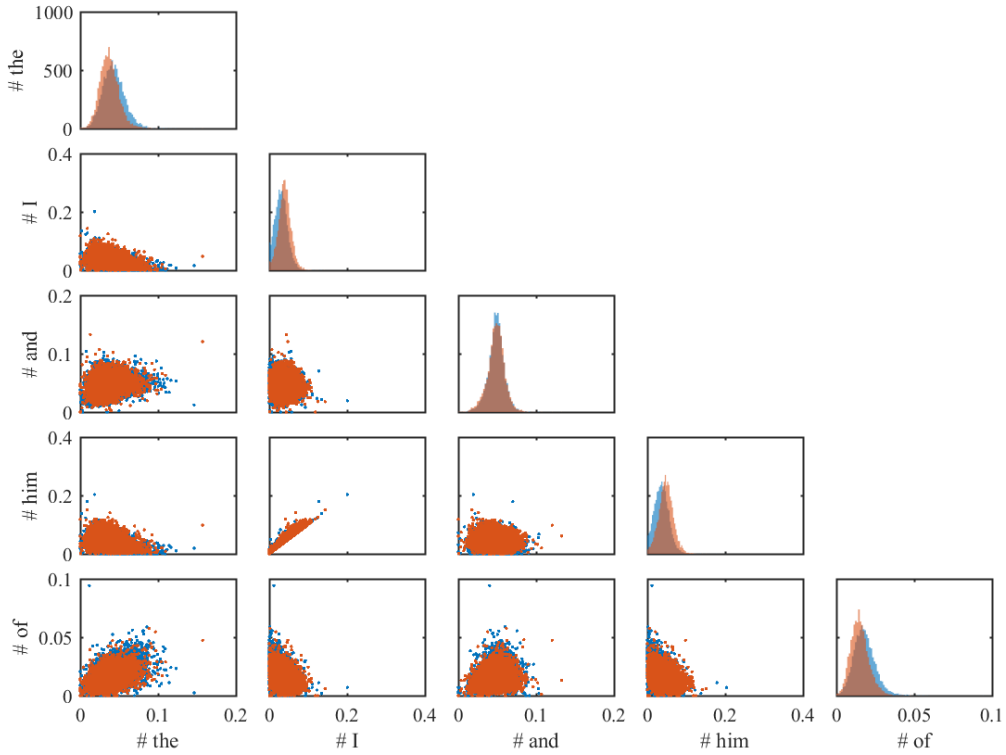


Figure 2: Corner plot of the predictors 9-13 from above. Red indicates female, and blue indicates male. The diagonal shows the histogram of each predictor, whereas the plot (i,j) is the correlation between predictor i and j.

4 Methods

There are various methods within machine learning, but in this project we will compare two methods, namely decision trees with adaptive boosting (AdaBoost) and support vector machines (SVM).

4.1 Decision trees with adaptive boosting

Boosted decision trees is a variant of the decision tree algorithm, which uses an iterative method to account for misclassification of the previous iteration. The first "generation" of the boosted decision tree is just a regular decision tree; the "boosting" in the algorithm comes by re-weighing the data-points after the decision tree algorithm has been applied. The data-points are re-weighted such that the misclassified points are given extra weight (the weights of correctly-classified points do not change), and this extra weight makes it more likely that the decision tree algorithm, when run again, will correctly label these re-weighted points (at the cost of now miss-classifying other points). Each iteration of this algorithm is stored as a generation of the tree, and by repeating the steps the boosted decision tree is grown. The final classification is then made by taking the combined output of all generations; this output is a fuzzy-logical value for the probability that a given data-point has a given label, and the final classification step is then decided by the threshold value chosen by the user. In this case, as we are equally interested in the two classes, the threshold value is set to 50 %.
The adaptive boosting method describes a way of re-weighing misclassified points, and

a way of combining the outputs of each generation, but we will not describe this in detail, as the choice for the specific boosting method is somewhat arbitrary, and any other method, such as LogitBoost, could have been used.

4.2 Support vector machines

For this project we will also use a Support Vector Machine (SVM); specifically we will use a Soft-Margin SVM. which is a type of SVM which allow for some slack in the separation of data-points. The SVM works by detecting the best possible (hyper)plane that separates the data-points; in the simple case, where the data is linearly separable, this is just a regular (hyper)plane in feature space. However, if the data is not linearly separable one can instead use the so-called kernel methods to transform the data from the feature space to a higher dimensional space, use a (hyper)plane to separate the now linearly-separable data, and transform the data back into feature space, where the resulting decision boundary is not necessarily linear in this space. This allow for some highly complex decision boundaries which can (hopefully) separate the data convincingly.

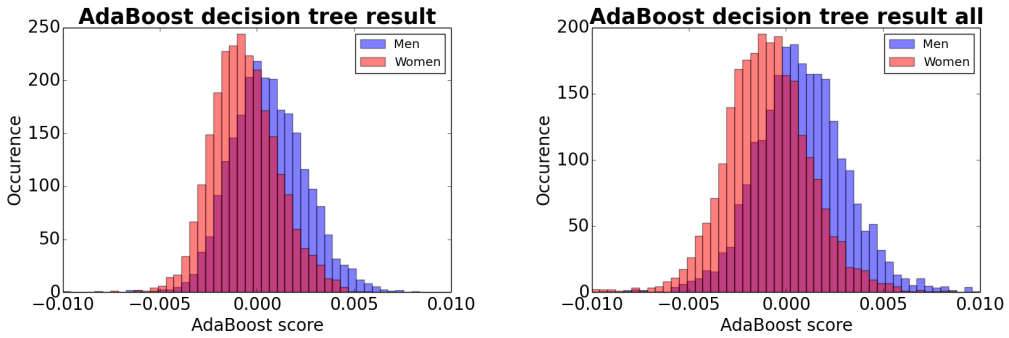
This method does come at a cost however, since we need to tune some of the parameters of the SVM to give the optimal results: The slack-parameter C and a kernel parameter σ. The slack parameter is a regularizer for how much we allow points to be misclassified, and the kernel parameter is (in this case) the width of the Gaussian kernel used to transform the data. There are other choices for the basis function of the transformation, but the Gaussian is easy and robust. The tuning of the parameters is done by five-fold cross-validation of the training data, and the set of values for C and σ which has the lowest cross-validation loss is chosen as the SVM parameters.

5 Results

Using the boosted decision trees with the AdaBoost method, we grow our decision tree using the first 8 predictors - that is we did not train the tree on the frequencies of the five test words. We use 50% of the data to grow the tree, and the remaining data for testing. The resulting separation of the test data by the decision tree is shown in figure 3a. The accuracy of the decision tree is 63.4%.
We also run the algorithm using all 13 predictors, and again we use 50% of the data to grow the tree. The results using these predictors is shown in figure 3b. The accuracy of the decision tree is here 65.8%.
We now look at the importance of each of the predictors in both cases. This is shown in figure 4. Note that the frequency of three of the five test words rank highly when using all 13 predictors, suggesting that some contextual linguistic features in the data set are more important many of more descriptive ones (such as average word length etc.).

For this project we also trained an SVM for comparison with the AdaBoost method. Here we used the predictors 1-8. With these 8 predictors, we train the SVM on 20% of the data, and use the remaining 80% of the data as test data. The 5-fold cross-validation error on the training sample is shown in figure 5a; the smallest cross-validation error is found for $(C, \sigma) = (10^2, 10)$. The scores of the test data generated by the SVM is shown in figure 6a, with an accuracy on the test data of 62.4%.

To illustrate the role of the division of training and test data, we train the SVM again now using 50% of the data as training data. The associated 5-fold cross validation error plot



(a) The separation of the test data using the AdaBoost method and predictors 1-8. This yields an accuracy of 63.4%.

(b) The separation of the test data using the AdaBoost method and predictors 1-13. This yields an accuracy of 65.8%.

Figure 3: The tree is grown using 50% of the data set and remaining 50% is the test data. A value smaller (larger) than zero corresponds to the label "female" ("male"). The colors of the two histogram corresponds to the true label.

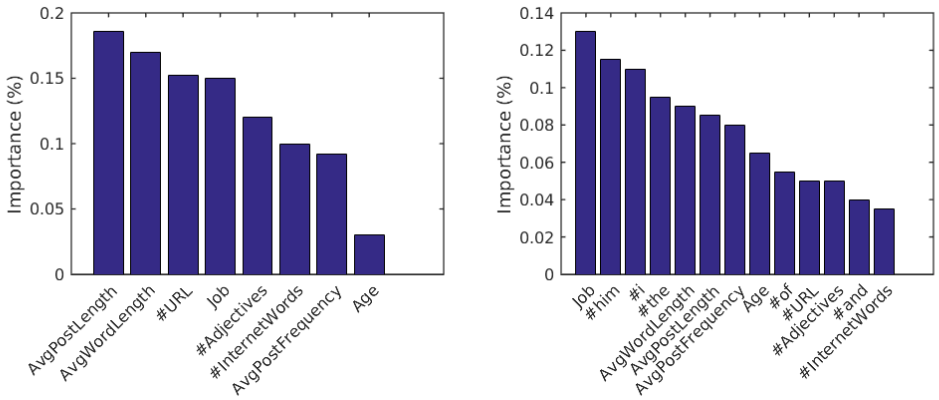
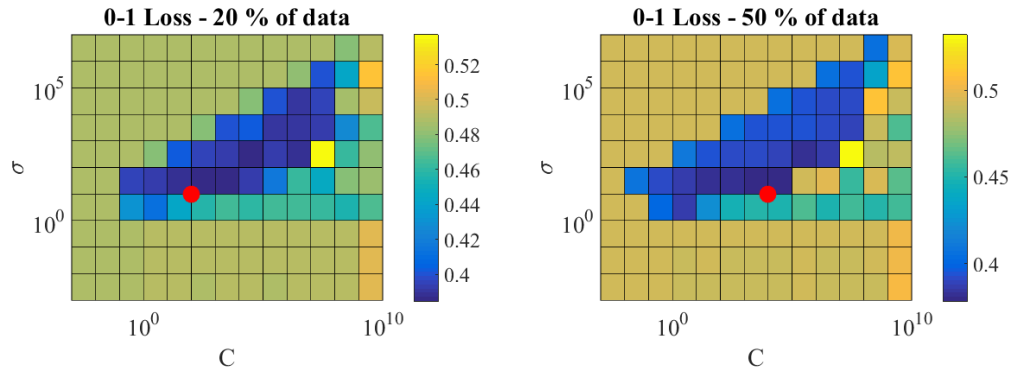


Figure 4: The importance of each of the predictors. On the left, we use the first 8 predictors, and on the right we use all 13 predictors.

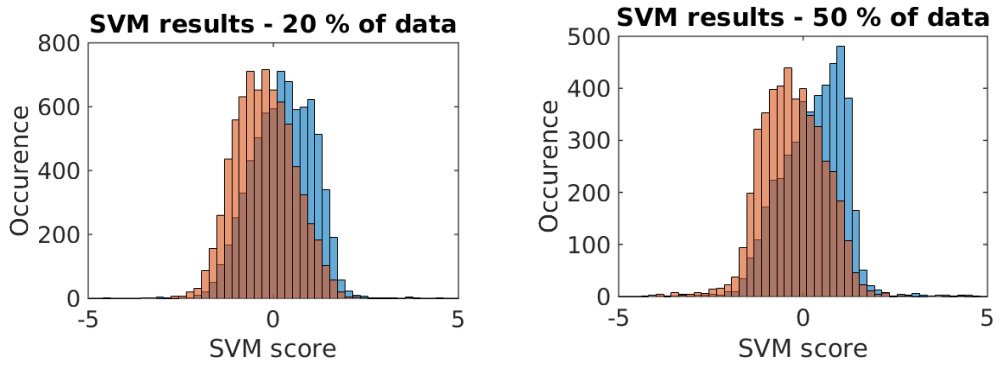
is shown in figure 5b; now with the lowest cross-validation error for $(C, \sigma) = (10, 10^4)$. The SVM score is shown in figure 6b; the increase in training data does only improves the performance by about one percent, as the accuracy on the test data is now 63.5%.



(a) Using 20% of the data as training data. Minimum loss found is 0.362 at (C,σ)=(100, 10).

(b) Using 50% of the data as training data. Minimum loss found is 0.350 at (C,σ)=(10, 10^4).

Figure 5: The five-fold cross-validation loss of the SVM as a function of the parameters C and σ. The red point shows the minimum loss, and the corresponding values of the parameters are used for the final SVM



(a) Using 20% of the data as training data. This yields an accuracy of 62.4%.

(b) Using 50% of the data as training data. This yields an accuracy of 63.5%.

Figure 6: The separation of the two genders in terms of their SVM score using predictors 1-8. A value smaller (larger) than zero corresponds to the label "female" ("male").

6 Discussion

For both the SVM and the AdaBoost using predictors 1-8 the accuracy is found to be around 63.4% - 63.5%, though the shape of the histograms in figures 3a and 6b is not completely alike. That the two algorithms agree to closely is hardly surprising, but it is satisfying that both yield the same results. There is however a clear runtime difference between SVM and AdaBoost: SVM requires extensive tuning, which has a rather large runtime on the order of 16 hours in our laptops, with AdaBoost being significantly faster as it requires no tuning of parameters - thus AdaBoost is also parameter-free, another advantage.

The best prediction accuracy we obtain is $\sim 66\%$, which is hardly impressive when compared to the best results from the articles we have read, which can come close to 89% [2]. However, the large difference seems to come from the fact that their predictors have been based largely on clever linguistic choices (writing styles parametrized in some clever way, etc.), which we do not have the linguistic knowledge to repeat.

Also interesting is the fact that the SVM's performance does not increase significantly with more training data after 20%. This suggest that the generalization potential has been reached, meaning that the algorithm does not become significantly better at generalizing as it gets more data. We saw a similar effect in the AdaBoost algorithm (an increase of about one percent), but we have not shown it in this project.

Using all 13 predictors for the AdaBoost algorithm increases the accuracy to 65.8%, a slight improvement of 2.4%. That such clear seperation in 5 very common words exists (as shown in figure 2) is somewhat surprising, but may reflect on the formal vs contextual writing style which is apparently distinctive in the two genders [1]. Investigating the parameter space for all 13 predictors with the SVM proved too big a runtime challenge for the deadline of this project.

7 Conclusion

We have used the machine learning algorithms boosted trees with adaptive boosting (adaBoost) and support vector machines (SVM) to attempt a gender classification on a dataset of more than 15,000 blogs. Using 8 predictors (defined in section 3) the two algorithms yield nearly the same separation of data with an accuracy of 63.4% for adaBoost and 63.5% for SVM, indicating that the two algorithms perform equally well on our data, though SVM has a significantly larger runtime. An additional training has been run using all 13 predictors using adaBoost, which improved the accuracy up to 65.8%. These rates are significantly better than blind guesses, though not nearly as well as the best in literature [2].

References

[1] J. Schler, M. Koppel, S. Argamon & J. Pennebaker, *Effects of Age and Gender on Blogging*, , Proc. of the Association for the Advancement of Artificial Intelligance (March 2006)

[2] A. Mukherjee & B. Liu, *Improving Gender Classification of Blog Authors*, Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing, p. 207217, 2010

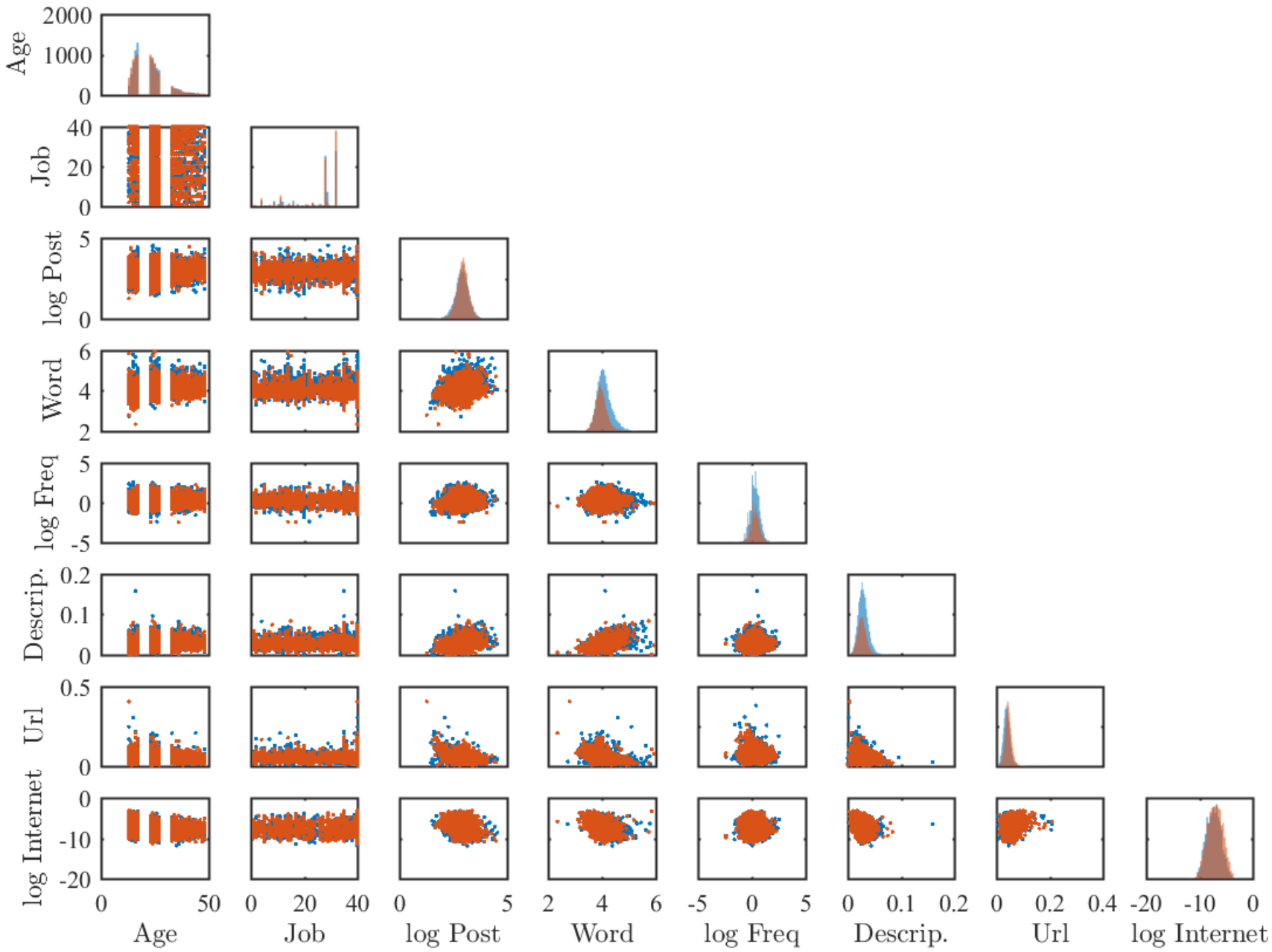


Figure 7: Corner plot of the predictors 1-8 from above. Red indicates female, and blue indicates male. The diagonal shows the histogram of each predictor, whereas the plot (i,j) is the correlation between predictor i and j.